School of Data and Computer Science, Sun Yat-sen University

E07 FF Planner

16337102 Zilin Huang

October 19, 2018

Contents

1 Examples

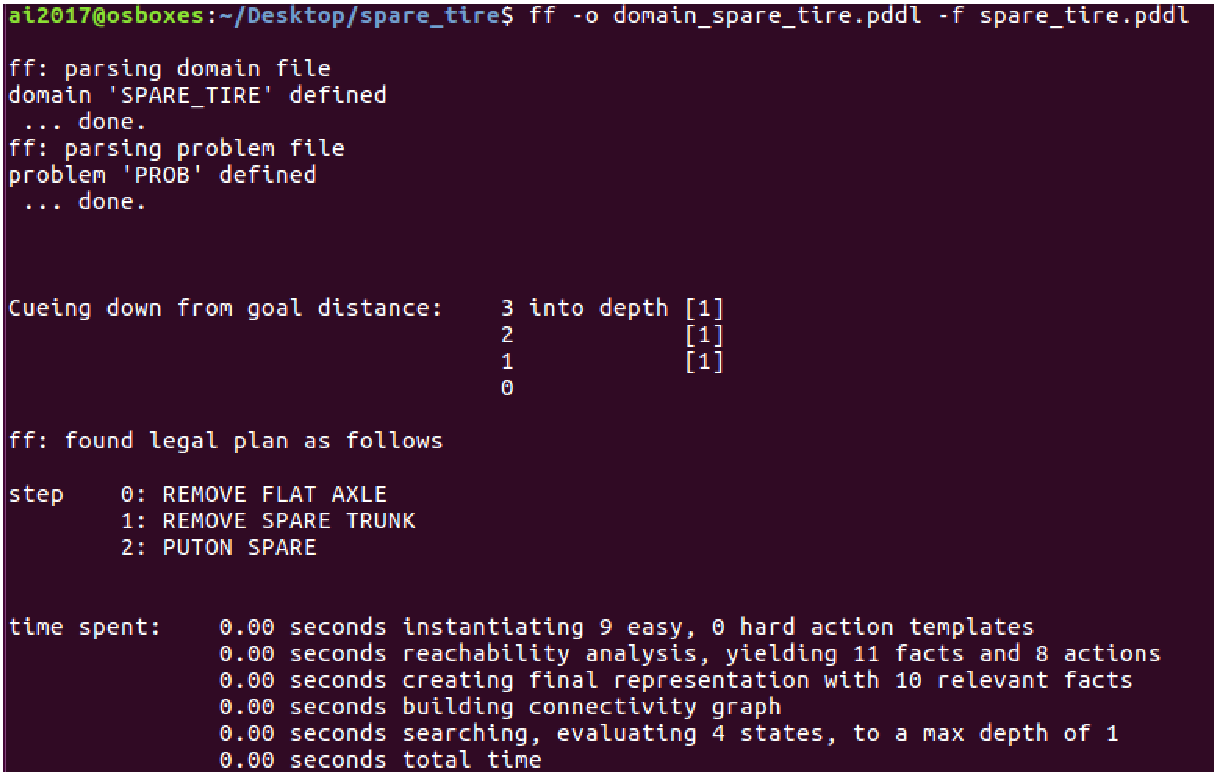
1.1 Spare Tire

domain_spare_tire.pddl

```
(define (domain spare_tire)
  (:requirements :strips :equality:typing)
  (:types physob location)
  (:predicates  (Tire ?x - physob)
                (at ?x - physob ?y - location))

(:action Remove
             :parameters (?x - physob ?y - location)
             :precondition (At ?x ?y)
             :effect (and (not (At ?x ?y)) (At ?x Ground)))

  (:action PutOn
             :parameters (?x - physob)
             :precondition (and (Tire ?x) (At ?x Ground)
                                (not (At Flat Axle)))
             :effect (and (not (At ?x Ground)) (At ?x Axle)))
  (:action LeaveOvernight
             :effect (and (not (At Spare Ground)) (not (At Spare Axle))
                          (not (At Spare Trunk)) (not (At Flat Ground))
                          (not (At Flat Axle)) (not (At Flat Trunk)) ))
 )
```

spare_tire.pddl

```
(define (problem prob)
 (:domain spare_tire)
 (:objects Flat Spare -physob Axle Trunk Ground - location)
 (:init (Tire Flat)(Tire Spare)(At Flat Axle)(At Spare Trunk))
 (:goal (At Spare Axle))
)
```



```
ai2017@osboxes:~/Desktop/spare_tire$ ff -o domain_spare_tire.pddl -f spare_tire.pddl

ff: parsing domain file
domain 'SPARE_TIRE' defined
 ... done.
ff: parsing problem file
problem 'PROB' defined
 ... done.



Cueing down from goal distance:    3 into depth [1]
                                   2            [1]
                                   1            [1]
                                   0

ff: found legal plan as follows

step    0: REMOVE FLAT AXLE
        1: REMOVE SPARE TRUNK
        2: PUTON SPARE


time spent:    0.00 seconds instantiating 9 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 11 facts and 8 actions
               0.00 seconds creating final representation with 10 relevant facts
               0.00 seconds building connectivity graph
               0.00 seconds searching, evaluating 4 states, to a max depth of 1
               0.00 seconds total time
```

1.2 Briefcase World

Please refer to `pddl.pdf` at page 2. Please pay More attention to the usages of `forall` and `when`.

For more examples, please refer to `ff-domains.tgz` and `benchmarksV1.1.zip`. For more usages of FF planner, please refer to the documentation `pddl.pdf`.

2 Tasks

2.1 8-puzzle

1	2	3
7	8	
6	4	5

Please complete `domain_puzzle.pddl` and `puzzle.pddl` to solve the 8-puzzle problem.

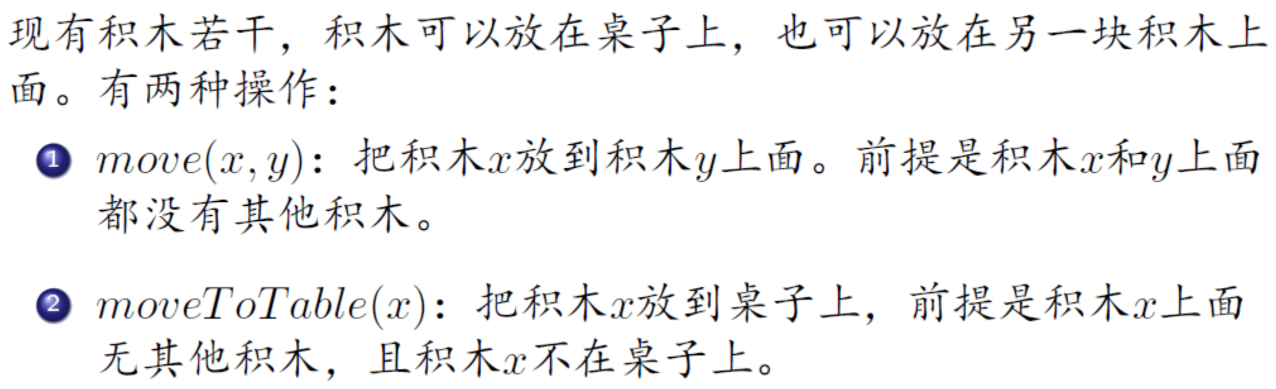
domain_puzzle.pddl

```
(define (domain puzzle)
  (:requirements :strips :equality:typing)
  (:types num loc)
  (:predicates ())

(:action slide
             :parameters ()
             :precondition ()
             :effect ()
 )
)
```

domain_puzzle.pddl

```
(define (problem prob)
 (:domain puzzle)
 (:objects )
 (:init )
 (:goal ())
)
```

2.2 Blocks World



现有积木若干，积木可以放在桌子上，也可以放在另一块积木上面。有两种操作：

1. $move(x, y)$：把积木x放到积木y上面。前提是积木x和y上面都没有其他积木。
2. $moveToTable(x)$：把积木x放到桌子上，前提是积木x上面无其他积木，且积木x不在桌子上。

Please complete the file `domain_blocks.pddl` to solve the blocks world problem. You should know the usages of `forall` and `when`.

domain_blocks.pddl

```
(define (domain blocks)
  (:requirements :strips :typing:equality
                 :universal-preconditions
                 :conditional-effects)
  (:types physob)
  (:predicates
            (ontable ?x - physob)
            (clear ?x - physob)
            (on ?x ?y - physob))

  (:action move
             :parameters (?x ?y - physob)
             :precondition ()
             :effect ()
             )

  (:action moveToTable
             :parameters (?x - physob)
             :precondition ()
             :effect ( )
 )
```

blocks.pddl

```
(define (problem prob)
 (:domain blocks)
 (:objects A B C D E F - physob)
 (:init (clear A)(on A B)(on B C)(ontable C) (ontable D)
  (ontable F)(on E D)(clear E)(clear F)
)
 (:goal  (and (clear F) (on F A) (on A C) (ontable C)(clear E) (on E B)
         (on B D) (ontable D)) )
 )
```

Please submit a file named E07_YourNumber.pdf, and send it to ai_2018@foxmail.com

3 Codes

domain_puzzle.pddl

```
(define (domain puzzle)
    (:requirements :strips :equality :typing)
    (:types num loc)
    (:constants B - num)
    (:predicates (adjacent ?x - loc ?y - loc)
                 (at ?x - num ?y - loc))

(:action slide
                :parameters (?t - num ?x - loc ?y - loc)
                :precondition (and (at ?t ?x) (adjacent ?x ?y)
                                   (at B ?y))
                :effect (and (at B ?x) (at ?t ?y)
                             (not (at ?t ?x)) (not (at B ?y)) )
)
)
```

puzzle.pddl

```
(define (problem prob)
        (:domain puzzle)
        (:objects num1 num2 num3 num4 num5 num6 num7 num8 - num
                  P1 P2 P3 P4 P5 P6 P7 P8 P9 - loc)
        (:init (at num1 P1) (at num2 P2) (at num3 P3) (at num7 P4)
               (at num8 P5) (at B P6) (at num6 P7) (at num4 P8)
               (at num5 P9)
               (adjacent P1 P2) (adjacent P1 P4)
               (adjacent P2 P1) (adjacent P2 P3)
               (adjacent P2 P5) (adjacent P3 P2)
               (adjacent P3 P6) (adjacent P4 P1)
```

```
                      (adjacent P4 P5) (adjacent P4 P7)
                      (adjacent P5 P2) (adjacent P5 P4)
                      (adjacent P5 P6) (adjacent P5 P8)
                      (adjacent P6 P3) (adjacent P6 P5)
                      (adjacent P6 P9) (adjacent P7 P4)
                      (adjacent P7 P8) (adjacent P8 P5)
                      (adjacent P8 P7) (adjacent P8 P9)
                      (adjacent P9 P6) (adjacent P9 P8))
            (:goal (and (at num1 P1) (at num2 P2) (at num3 P3) (at num4 P4)
                      (at num5 P5) (at num6 P6) (at num7 P7) (at num8 P8)
                      (at B P9)) )
)
```

domain_blocks.pddl

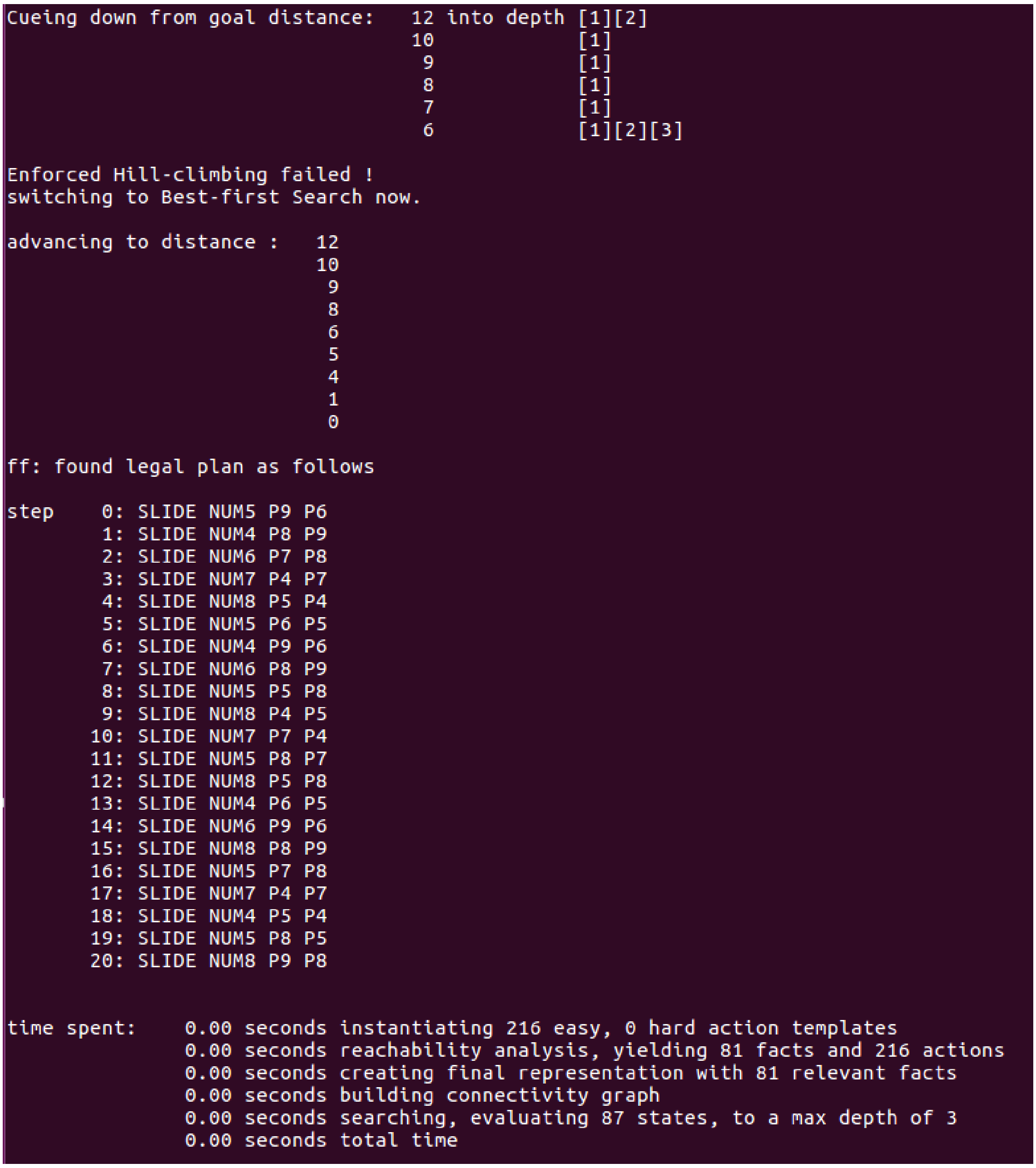
```
(define (domain blocks)
            (:requirements :strips :typing :equality
                                                    :universal-preconditions
                                                    :conditional-effects )
      (:types physob)
      (:predicates
                        (ontable ?x - physob)
                        (clear ?x - physob)
                        (on ?x ?y - physob)
                        )
      (:action move
                        :parameters (?x ?y - physob)
                        :precondition (and (clear ?y) (clear ?x))
                        :effect ( and (not (clear ?y)) (on ?x ?y)
                                            (forall (?z)
                                                      (when (on ?x ?z)
                                                      (and (not (on ?x ?z)) (clear ?z)) ))
                                            (when (ontable ?x) (not (ontable ?x)) )  )
                        )
```

```
    (:action moveToTable
                :parameters (?x - physob)
                :precondition (and (clear ?x) (not (ontable ?x)))
                :effect (and (ontable ?x)
                             (forall (?z)
                                    (when (on ?x ?z)
                                    (and (not (on ?x ?z)) (clear ?z)) ))
                )
)
)
```

blocks.pddl

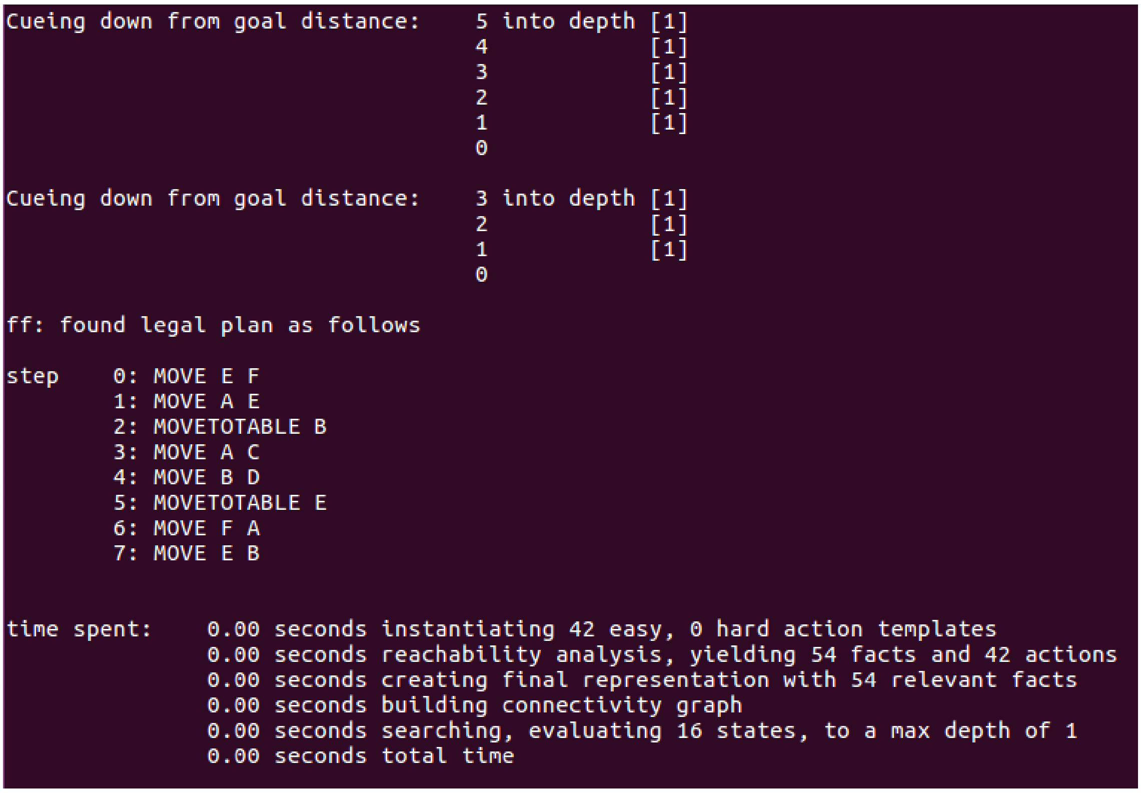
```
(define (problem prob)
        (:domain blocks)
        (:objects A B C D E F - physob)
        (:init (clear A) (on A B) (on B C) (ontable C) (ontable D)
               (ontable F) (on E D) (clear E) (clear F) )
        (:goal (and (clear F) (on F A) (on A C) (ontable C)
               (clear E) (on E B) (on B D) (ontable D) )
)
)
```

4 Results



```
Cueing down from goal distance:   12 into depth [1][2]
                                  10            [1]
                                   9            [1]
                                   8            [1]
                                   7            [1]
                                   6            [1][2][3]

Enforced Hill-climbing failed !
switching to Best-first Search now.

advancing to distance :   12
                          10
                           9
                           8
                           6
                           5
                           4
                           1
                           0

ff: found legal plan as follows

step    0: SLIDE NUM5 P9 P6
        1: SLIDE NUM4 P8 P9
        2: SLIDE NUM6 P7 P8
        3: SLIDE NUM7 P4 P7
        4: SLIDE NUM8 P5 P4
        5: SLIDE NUM5 P6 P5
        6: SLIDE NUM4 P9 P6
        7: SLIDE NUM6 P8 P9
        8: SLIDE NUM5 P5 P8
        9: SLIDE NUM8 P4 P5
       10: SLIDE NUM7 P7 P4
       11: SLIDE NUM5 P8 P7
       12: SLIDE NUM8 P5 P8
       13: SLIDE NUM4 P6 P5
       14: SLIDE NUM6 P9 P6
       15: SLIDE NUM8 P8 P9
       16: SLIDE NUM5 P7 P8
       17: SLIDE NUM7 P4 P7
       18: SLIDE NUM4 P5 P4
       19: SLIDE NUM5 P8 P5
       20: SLIDE NUM8 P9 P8


time spent:    0.00 seconds instantiating 216 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 81 facts and 216 actions
               0.00 seconds creating final representation with 81 relevant facts
               0.00 seconds building connectivity graph
               0.00 seconds searching, evaluating 87 states, to a max depth of 3
               0.00 seconds total time
```

Figure 1: Pzzle



```
Cueing down from goal distance:    5 into depth [1]
                                   4            [1]
                                   3            [1]
                                   2            [1]
                                   1            [1]
                                   0

Cueing down from goal distance:    3 into depth [1]
                                   2            [1]
                                   1            [1]
                                   0

ff: found legal plan as follows

step    0: MOVE E F
        1: MOVE A E
        2: MOVETOTABLE B
        3: MOVE A C
        4: MOVE B D
        5: MOVETOTABLE E
        6: MOVE F A
        7: MOVE E B


time spent:    0.00 seconds instantiating 42 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 54 facts and 42 actions
               0.00 seconds creating final representation with 54 relevant facts
               0.00 seconds building connectivity graph
               0.00 seconds searching, evaluating 16 states, to a max depth of 1
               0.00 seconds total time
```

Figure 2: Blocks